Data structures champ_no_s and cham_elem_s

Summarized:

This document describes the data structures cham_no_s and cham_elem_s.

One also gives the list of the principal utilities working on these data structures.

Two new SD are defined: cham_no_s and cham_elem_s which contain same information as the SD cham_no and cham_elem but which is more “simple” to handle in FORTRAN.

There exist utilities making it possible to transform a cham_no into cham_no_s (and reciprocally) (in the same way for the cham_elem).

These SD will be thus in general temporary SD making it possible to work more simply.

Notice important:

The SD cham_no_s and cham_elem_s are not as general as the SD cham_no and cham_elem. For the cham_no_s, one describes only the fields carried by the nodes of the mesh (and not the possible late nodes), For the cham_elem_s, one describes only the fields carried by the finite elements whose mesh support is a mesh of the mesh (and not a late mesh)

Contents

1 SD cham_no_s

```
cham_no_s (K19)          :: = record
```

- ".CNSK" : OJB S V K8 length = 2
- ".CNSD" : OJB S V I long = 2
- ".CNSC" : OJB S V K8 length = nb_CMP
- ".CNSV" : Long OJB S V R/C/I/ … = nb_NOEUD*nb_CMP
- ".CNSL" : OJB S V L long = nb_NOEUD*nb_CMP

1.1 Contained OJB

This SD is used to describe a field of variables carried by the nodes of a mesh.

1.2 Object ".CNSK"

".CNSK" (1)	netted : name of the mesh subjacent with the cham_no_s.
".CNSK" (2)	nomgd : name of the quantity associated with the cham_no_s ("DEPL_R", "SIEF_R",...)

1.3 Object ".CNSD"

".CNSD" (1)	nb_NOEUD : many nodes of the subjacent mesh.
".CNSD" (2)	nb_CMP : maximum number of the CMPS carried by the nodes.

1.4 Object ".CNSC"

".CNSC" (ICMP)	cmp_i : name of the ième CMP of the cham_no_s

Note::
The order of the CMPS in .CNSC can be unspecified. One is not obliged to respect the order of the catalog of quantities. On the other hand, the CMPS must be part of the CMPS of the quantity nomgd.

1.5 Object ".CNSV"

This object contains the values of the cham_no_s. The type of this vector JEVEUX (R/C/I/K8,…) is that of the quantity nomgd. Its dimension is nb_NOEUD*nb_CMP; i.e. all the nodes of mesh can carry all the CMPS described in .CNSC.

One reaches NOEUD ICMP-ème CMP of INO-ème by the formula:

```
VALEUR (INO, ICMP) = .CNSV ((INO-1) *nb_CMP + ICMP)
```

Note::
The presence (or the absence) of a CMP on NOEUD is indicated via object .CNSL (see below). During the creation of a cham_no_s, its nonaffected values are in settings with "undef" for better detecting their illicit use.

1.6 Object ".CNSL"

This object contains the Boolean ones indicating the presence (or the absence) of the values of the cham_no_s.
Its dimension is nb_NOEUD*nb_CMP; one moves there in the same way that in object .CNSV

One NOEUD examines the presence of ICMP-ème CMP of INO-ème by the formula:

```
EXIST (INO, ICMP) = .CNSL ((INO-1) *nb_CMP + ICMP)
```

2 SD cham_elem_s

2.1 Description of the SD

This data structure allows to represent the values of the fields discretized on meshes of a mesh.

More precisely, the access to an actual value (or complex,…) field is done while specifying:

- the number of the mesh supporting the finite element (IMA),
- the number of the point in mesh (IPT),
- the number of the subpoint in point (ISP) (ISP =1 in general),
- the number of the component of the quantity associated with field (ICMP),

```
cham_elem_s (K19)                :: = record

     ♦  ".CESK"  : OJB   S V K8              length = 3
     ♦  ".CESD"  : OJB   S V I               long = 5 + 4*nb_MAILLE
     ♦  ".CESC"  : OJB   S V K8              length = nb_CMP
     ♦  ".CESV"  : OJB   S V /R/C/I/..      length = nbval
     ♦  ".CESL"  : OJB   S V L               long = nbval
```

2.2 Object .CESK

.CESK (1)	netted : name of the mesh subjacent with the cham_elem_s.
.CESK (2)	nomgd : name of the quantity associated with the cham_elem_s ("DEPL_R", "SIEF_R",...)
.CESK (3)	/"ELNO": field known with the nodes of the elements, /"ELGA": field known with Gauss points of the elements, /"ELEM": constant field by element (it will be said whereas it is known at the center of gravity)

2.3 Object .CESD

.CESD (1)	nb_MAILLE : number of meshes of the subjacent mesh.
.CESD (2)	nb_CMP : many CMPS carried by the points. It is the dimension of object .CESC
.CESD (3)	nbptmx : maximum amongst points carried by meshes
the .CESD (4)	nbspmx : maximum amongst subpoints carried by the points of meshes
the .CESD (5)	nucmpmx : the highest sequence number of the possible CMP of cham_elem_s (in the order of object .CESC)
.CESD (5+4* (ima-1) +1)	nbpt (ima): many points of the mesh ima.
.CESD (5+4* (ima-1) +2)	nbsp (ima): number of subpoints of the mesh ima.
.CESD (5+4* (ima-1) +3)	nbcmp (ima): maximum number of the CMPS carried by the subpoints of the points of the mesh ima.
.CESD (5+4* (ima-1) +4)	IAD (ima): IAD+1 is the address in objects .CESL and .CESV of the 1st CMP of the 1st subpoint of the 1st point of the mesh ima (if they exist)

2.4 Object .CESC

.CESC (ICMP)	cmp_i : name of the ième CMP of the cham_elem_s

Note::

The order of the CMPS in .CESC can be unspecified. One is not obliged to respect the order of the catalog of quantities. On the other hand, the CMPS must be part of the CMPS of the quantity nomgd (except quantity VARI_R).

2.5 Objects .CESL and .CESV

These objects contain the values of the cham_elem_s (.CESV) and Boolean (.CESL) indicating if these values were affected (or if they are undetermined).

Type JEVEUX (R/C/I/K8,…) object .CESV is that of the quantity nomgd.

The dimension of these 2 vectors is nbval :

nbval is the sum on all meshes the ima of nbpt (ima) * nbsp (ima) *nbcmp (ima)

to reach the ICMP - ème CMP of the ISP - ème subpoint of the IPT - ème POINT of the IMA - ème MESH of a cham_elem_s , one uses utility routine CESEXI :

```
CAL CESEXI (STOP, JCESD, JCESL, IMA, IPT, ISP, ICMP, IAD)
```

where: JCESD and JCESL are the addresses of objects .CESD and .CESL of the cham_elem_s. IAD is the “output” of this routine.

if IAD > 0 , that wants to say that the required component exists in the cham_elem_s. One can then recover it by: VALEUR = ZR (JCESV-1+IAD) (if the field is real).

if IAD < 0 , that wants to say that the required component has a possible core in the cham_elem_s but that it not affected currently. One can then affect a value in the cham_elem_s while making:

```
ZR (JCESV-1+IAD) = VALEUR
ZR (JCESL-1+IAD) = .TRUE
```

if IAD = 0 , that wants to say that the required component does not have possible room in the cham_elem_s. I.e. that one at least of the following conditions is checked:

```
IMA > nb_MAILLE
IPT > nbpt (IMA)
ISP > nbSP (IMA)
ICMP > nbcmp (IMA)
```

2.6 Example of loop on the values of a cham_elem_s

```
CAL JEVEUO (THESE ".CESD", "It, JCESD)
CAL JEVEUO (THESE ".CESL", "It, JCESL)
CAL JEVEUO (THESE ".CESV", "It, JCESV)
NBMA = ZI (JCESD-1+1)
C 40, IMA = 1, NBMA
   NBPT = ZI (JCESD-1+5+4* (IMA-1) +1)
   NBSP = ZI (JCESD-1+5+4* (IMA-1) +2)
   NBCMP = ZI (JCESD-1+5+4* (IMA-1) +3)
   C 30, IPT = 1,NBPT
      C 20, ISP = 1, NBSP
         C 10, ICMP = 1, NBCMP
            CAL CESEXI (STOP, JCESD, JCESL, IMA, IPT, ISP, ICMP, IAD)
            IF (IAD.GT.0)  VALEUR = ZR (JCESV-1+IAD)
```

3 utility Routines

.

CARCES	to transform a card into a cham_elem_s
CELCES	to transform a cham_elem into cham_elem
CESCES	to change the discretization of a cham_elem (ELNO/CART/ELGA)
CESCNS	to transform a cham_elem_s into a cham_no_s
CESCRE	to create a cham_elem_s
CESEXI	to test the existence of a CMP of a point of a mesh of a cham_elem_s
CESRED	"to reduce" a cham_elem_s on a list of meshes and/or a list of CMPS.
CESTAS	"retasser" contents of a cham_elem_s
CNOCNS	to transform a cham_no into cham_no_s
CNSCES	to transform a cham_no_s into cham_elem
CNSCNO	to transform a cham_no_s into cham_no
CNSCRE	to create a cham_no_s
CNSPRJ	to project a cham_no_s on another mesh
CNSRED	"nodes list to reduce" a cham_no_s on one and/or a list of CMPS.

COPISD	to copy a cham_no_s or a cham_elem_s
DETRSD	to destroy a cham_no_s or a cham_elem_s
IMPRSD	to print from listing a cham_no_s or a cham_elem_s